A Heuristic for Paradigms

Joseph E. Grimes
Cornell University and
Summer Institute of Linguistics

This paper helps clarify one of the pervasive problems of linguistic analysis: the interaction between the paradigmatic and syntagmatic dimensions of language. Paradigms are sets of alternatives: the speaker must decide on one member of the set to use, and the hearer must figure out which he used. In a syntagm or construction, an element chosen out of one paradigm is put together with elements chosen out of others. Thus far all grammars of all languages agree.

The problem comes when we put the grammar together. The choices available in one paradigm turn out often to be limited by those made in some other paradigm that is part of the same construction. Grammar is never as simple as a Cartesian product of paradigms.

Various forms of grammar have various means, none of them quite satisfying, to express these limitations. A common one is footnotes about irregularities; ad hoc features to trigger or block special rules when needed are also used.

Grammar ought to highlight the mutual constraints between paradigms and constructions, not downplay them. Halliday's systemic grammar has done well in this regard (Halliday 1961, Hudson 1971). It is already known to computational linguists through Winograd's work (1972). The heuristic, based on work by Lowe, Dooley, and myself (in press), is expressed within Halliday's framework here, though it is applicable within any other model of language as well.

In systemic terms a paradigm is known as a 'system'. A choice in one system can be the entry condition for another system, one part of a system can have different properties of combination from another part, and two or more systems can be activated together as the basis for a construction. The heuristic is intended to clarify something that is more often guessed at than proved: what element belongs to what system.

What I find, on looking at languages other than English, is that membership in a Hallidayan system is by no means obvious in all cases. This is true for two reasons: first, some elements have properties that permit us to assign them to more than one system, and second, some elements are artifacts of the mapping relation between systems and forms, rather than direct manifestations of choices within systems.

The Data

Table (1) gives some data which illustrate this general point by means of a limited example. It reports cooccurrences among a particularly complex subset of the prefixes to the verb in Huichol, a Uto-Aztecan language spoken in the Mexican Sierra Madre. A 1 in the table means that the prefix at the head of the column has been observed in the combination that the row reports. For this language there are exactly 15 observable combinations of these prefixes, each represented by one row in Table (1). The order in which the rows are written down makes no difference, nor does the order in which the columns appear. ka1- and ka2- are homophonous forms that occupy different positions in the prefix string and have different meanings. & stands for a high back unrounded vowel.

(1)

ka1	ke	p&	m&	ka2	ni
1	0	1	0	1	0
1	0	1	0	0	0
1	0	0	0	1	1
1	0	0	0	0	1
0	1	0	0	0	1
0	1	0	0	0	0
0	0	1	0	1	0
0	0	1	0	0	0
0	0	0	1	1	1
0	0	0	1	1	0
0	0	0	1	0	1
0	0	0	1	0	0
0	0	0	0	1	0
0	0	0	0	0	1
0	0	0	0	0	0

The simple fact that two forms cannot cooccur with each other is the most obvious basis for saying that those two are members of a single system, that they are in opposition as alternatives in a paradigm, that the choice of one as over against the other has linguistic significance. In Table (1), for example, p&- does not occur in any combination where ni- occurs, and vice versa.

Noncooccurrence patterns

The patterns of noncooccurrence are derived from Table (1) by a simple algorithm:

```
For each column:
    Create a vector of as many 0's as
     there are columns
    For each row that has a 1 in the
     column in question:
        Unite that row with the
         vector.
Complement the vector.
```

Each of the uncomplemented vectors represents the union of all the combinations which the form at the head of its column enters into. The 1's in its complement therefore identify the elements with which it cannot cooccur.

The Huichol data -- and this is true of other languages, possibly of all languages -- do not allow us to draw immediate conclusions about mutual exclusiveness or simple comembership in systems. The prefixes represented by the complement vectors of each form are

(2) kal: ke, m&
    ke: kal, p&, m&, ka2
    p&: ke, m&, ni
    m&: kal, ke, p&
    ka2: ke
    ni: p&

A form like p&- can be assigned to one system in opposition with ni-, and to another in opposition with ke- and m&-; but kali-, which could also go into a system with ke and m&-, cooccurs with p&- and therefore cannot represent an alternative to it. The logic of systems in grammar is more complex than independent commutation, with the Cartesian products that that implies, in which each form of one set cooccurs with every form of another.

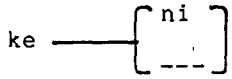
Decomposition

The true interdependency of a systemic network can be captured in a cooccurrence graph by first decomposing Table (1). The most manageable decomposition strategy found so far is to start with the column that minimizes the number of 1's that would be removed from the table if all the rows that have 1's in that column were removed. We convert those rows into a component subgraph, then continue recursively on the table minus those rows until no rows are left, or until the zero row is left; then we also convert the zero row if there is one into a component subgraph. In the final step of the heuristic, the component subgraphs are united to give the complete cooccurrence graph. That graph of forms is the aim of the heuristic. It is not a systemic network diagram itself, but is rather a statement of a major constraint on the semantic systemic diagram that accounts for the forms.

Component subgraphs are formed by putting alternatives vertically in any order within square brackets, and connecting forms that cooccur in any order by horizontal lines. Absence of any form in a particular combination is represented by ---.

In Table (1) the two rows that contain 1's for ke- have a total of only three 1's in them; so those two rows are taken out for the first subgraph:

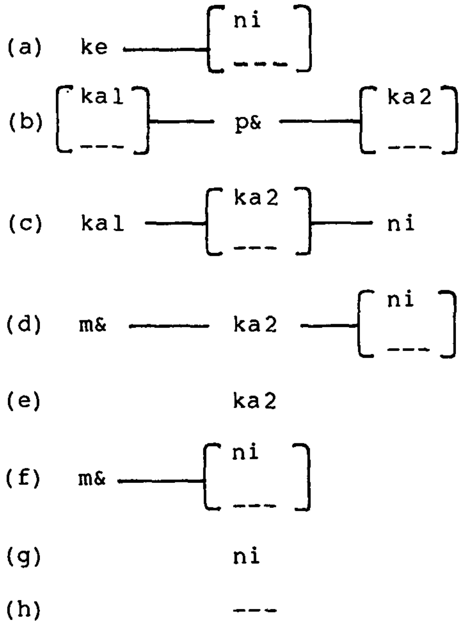
(3)



This subgraph, like the two rows of Table (1) that it represents, says that ke- can occur with or without ni-.

The full set of component subgraphs derived from Table (1) contains only simple alternatives and their Cartesian products:

(4)

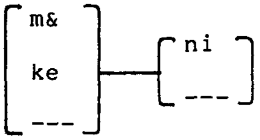


Union of component subgraphs

We unite these subgraphs by conflating what they have in common and symbolizing their differences as alternatives, by the distributive property. Four of the subgraphs, (a), (f), (g), and (h), can be combined without changing the picture of simple systems and

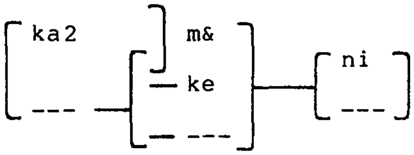
Cartesian products:

(5)



A restriction on Cartesian products appears, however, when we expand the composite diagram further. (d) has three out of four of its elements in common with elements already in the composite diagram (5). The fourth element, however, has nothing to do with ke- or its absence, but only with m&-. Here is where the discrepancies in noncooccurrence properties of different forms come into the picture, and here is where the Hallidayan device of linked brackets is needed in order to show up those discrepancies. The elements in (6) are reordered to disrupt the graphic shape given by (5) as little as possible:

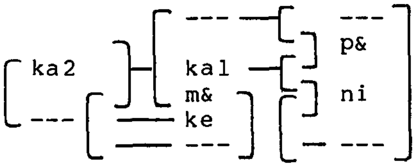
(6)



Cooccurrence graph

The complete cooccurrence graph is built up by continuing in the same way until all the component subgraphs are in it:

(7)



The use of two null symbols in a single set of alternatives does not mean that Huichol has two zero prefixes that contrast with each other, but rather that the graph is essentially nonplanar. Redundant nulls could be eliminated by crossing lines in an equivalent graph.

This diagram now shows all the constraints on cooccurrence that there are for these Huichol prefixes. It is not yet a systemic diagram, because systemic diagrams give differences in meaning and this one gives only cooccurrences of forms. The systemic diagram we come up with will, however, have to account for each of the constraints on cooccurrence given by this diagram.

Our scrutiny of cooccurrences and noncooccurrences has shown us what forms might be in opposition with each other in a semantic system, and how those forms interlock. That is as far as our explicit heuristic take us; but it narrows the field for semantic investigation considerably.

Computational aspects

Before I go on to show the payoff in terms of systems of meaningful choices, let me sketch the computational aspects of the heuristic. For a small problem like the one in the example, of course, no computing is needed. But were we to take in all 42 verb prefixes of Huichol, and state how they combine with suffixes and different stem types as well, the heuristic would never get off the ground with pencil and paper. It is a good example of how a computationally simple process, actually a twist on concordance generation, can bring order into an area where a linguist is otherwise all too likely to shrug his shoulders and define oversimplified systems, then write interminable footnotes about why they don't quite combine as he says they do.

A linguist in the field needs a three-step computational aid. Step One is data entry: take in occurring combinations of forms, which could as well be function words or suffixes or any combination of closed class phenomena, and develop a table like Table (1). Step Two is union: read the table and develop a vector for each form that shows the union of all its combinations. Step Three is decomposition: segregate out from the table the subsets of its rows that facilitate making its component subgraphs.

These three steps are easy to implement. The fourth step of the heuristic, forming the cooccurrence graph by uniting the component subgraphs, is at least an order of magnitude more complex, and may not be feasible for a small field computer.

Systemic diagram

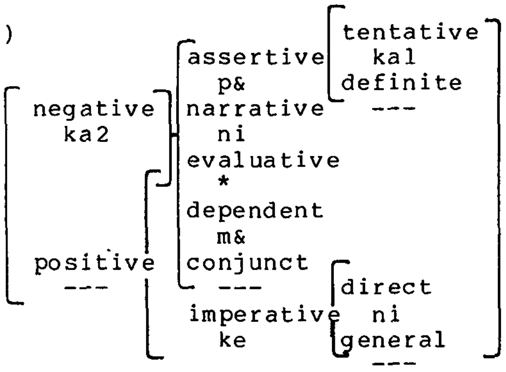
After the heuristic procedure is gone through, whether with pencil or by computer, the construction of a semantic hypothesis rich enough to account for all the patterns of cooccurrence can go ahead. This is a standard linguistic undertaking, and has two sides. The first is to investigate the reasons why one or another member of a noncooccurring set like the ones in (2) gets chosen. The reasons for choosing either member of a pair may not be the same in the context of one pattern of choices made in other systems as it is in other contexts. The second part of the semantic inquiry is to identify or combinations of forms whose presence is an artifact of the mapping between meaning and form, and not an assertion of a particular meaning.

This arbitrariness in the mapping relation shows up in two places in the example. When p&- is present, kal- has either a tentative or a very strong negative meaning: kaalp&míe means 'he might not go' or 'he shall not go!' (the meaning split is not too different from that of English terribly in terribly disfigured vs. terribly nice). With ni-, however, kal- has to be there when ka2-, the ordinary negative, is present, and may or may not be there when ka2- is absent. The requirement that kal- always go with ka2- in the presence of ni- eliminates the possibility of the two homophones ever being opposed to one another, with resulting confusion between negative and tentativn between negative and tentative meanings.

The other arbitrariness turns up on trying to relate m&- with ni-. m&- by itself is the sign of a dependent verb, and ni- by itself of an independent verb at a partice combination m&ni, however, has nothing to do with either of these meanings; it makes a statement of the speaker's opinion. I take it to be a morphologically complex expression of a separate term of the modal system.

Taking these discrepancies into account gives us a systemic diagram:

(8)



[narrative] implies kal
obligatory with [negative]
optional with [positive]
[evaluative] realized as m&+ni

It is by straightening out kal- and m&ni- that it becomes possible for us to give a systemic diagram plus a set of realization rules for it. The straightforward realization rules are written right into the diagram: for example, if you choose [negative], utter ka2-. The more complex realizations are given at the bottom of the diagram.

The terms of the systemic diagram are labels for semantic choices that have been explained elsewhere and do not concern us now; they do not constitute explanations in themselves. Once the arbitrary mappings are defined in realization rules, the diagram embodies only one real restriction on Cartesian products of paradigms, in that Huichol has no special negative imperative form. (It uses the negative declarative p&ka2 in its place.) The completeness of the analysis is supported by the fact that the interconnected paradigms of (8) have exactly 14 paths through them, and that together with the optional rule for the realization of kal- with ni-, these yield exactly the 15 rows of Table (1) with which we began.

Bibliography

Grimes, Joseph E., Ivan Lowe, and Robert A. Dooley. in press. Closed systems with complex restrictions. Anthropological Linguistics.

Halliday, Michael A. K. 1961. Categories of the theory of grammar. Word 17:241-292.

Hudson, R. A. 1971. English complex sentences. Amsterdam: North-Holland Publishing Company.

Winograd, Terry. 1972. Understanding natural language. New York: Academic Press.